Hierarchical Motion Decomposition for Dynamic Scene Parsing

Juan-Manuel Pérez-Rúa, Tomas Crivelli, Patrick Pérez, Patrick Bouthemy

HIERARCHICAL MOTION DECOMPOSITION FOR DYNAMIC SCENE PARSING

Juan-Manuel Pérez-Rúa[⋆†] *Tomas Crivelli*[⋆] *Patrick Pérez*[⋆] *Patrick Bouthemy*[†]

[⋆] Technicolor, Rennes - France
[†]Inria, Centre Rennes - Bretagne Atlantique - France

ABSTRACT

A number of applications in video analysis rely on a per-frame motion segmentation of the scene as key preprocessing step. Moreover, different settings in video production require extracting segmentation masks of multiple moving objects and object parts in a hierarchical fashion. In order to tackle this problem, we propose to analyze and exploit the compositional structure of scene motion to provide a segmentation which is not purely driven by local image information. Specifically, we leverage a hierarchical motion-based partition of the scene to capture a mid-level understanding of the dynamic video content. We present experimental results showing the strengths of this approach in comparison to current video segmentation approaches.

Index Terms— Video segmentation, motion models, conditional random fields, hierarchical motion decomposition.

1. INTRODUCTION AND RELATED WORK

In this work, we aim to recover the hierarchical motion organization of a scene given a pair of consecutive frames. The structure of the dynamic scene is tightly related to the natural hierarchical organization and layout of moving objects as illustrated in Fig.1. More specifically, our goal is to extract image regions and organize them according to motion hierarchies. The full visual motion of each region will be represented by the composition of incremental motions along a path of this hierarchy. Furthermore, the apparent scene motion induced by the camera intuitively lies at the top of the hierarchical structure that comprises the full scene motion decomposition, since it affects all the pixels. The resulting hierarchical visual motion representation captures view-based compositional characteristics of the scene motion.

Hierarchical motion decomposition of a scene is a process that can be useful for a number of applications in video processing. For instance, complex rotoscoping in video post-processing for the film industry requires to label by hand different independent moving objects and their moving parts. Current automatic solutions to this problem only focus on a single region and usually do not take into account mid-level motion dynamics of scenes [1, 2, 3]. Furthermore, one could think of new ways to propose spatio-temporal regions as candidates for action localization in video. For instance, Jain *et al.* [4] proposed to use an off-the-shelf segmentation method [5] to produce supervoxels that are later merged by criteria encompassing color, texture and motion information. The intermediate output of [4] is a tree-like set of spatio-temporal windows ("tubelets"), which is later used to feed bag-of-words classifiers to localize actions in video. However, hierarchical decomposition of video [4, 5, 6, 7, 8] does not usually manage to preserve intermediate semantic characteristics of the scene like delimitation of moving objects and parts, resulting in oversegmentation which is related to the complexities of color information in real imagery.

Fig. 1. Example of hierarchical motion representation of a dynamic scene. The full apparent motion of a point on the head of the bear is given by the composition of, in turn, the motion of the camera, of the bear body and of the bear head.

Our idea of motion decomposition comes from the biological vision research, where it is usually admitted that biological visual systems decompose scenes in common and relative motions [9]. A recent work on biological vision models this characteristic as a structured Bayesian vector analysis [10]. Scenes are decomposed according to the motion of nested spatial entities starting from global point-of-view changes, passing-by objects and object-members. The considered decomposition involves trees rooted on the camera motion. Due to practical limitations, the authors only model scenes as sets of points, and thus, cannot segment full images.

In the context of motion estimation from consecutive images, some works utilized multi-scale/multi-resolution techniques [11] to solve for long displacements by computing optical flow in an incremental way inside a coarse-to-fine strategy. In particular, for [12] this strategy is interpreted as a hierarchical optimization approach which can also be extended to perform motion segmentation. However, even though the per-

formed multi-scale modeling in [12] has an important value for the optimization, it remains difficult to interpret it as a hierarchical motion decomposition of the scene in the sense of [10].

The rest of the paper is organized as follows. In Section 2, we describe our method to obtain tree-structured motion decomposition of scenes from pairs of input images by stating the problem as a per-pixel label selection, where compositional motion models are chosen so to explain an instrumental optical flow field. In Section 3, we present experimental results that demonstrate the advantages of our method with respect to previous hierarchical approaches. Finally, we give concluding remarks in Section 4.

2. HIERARCHICAL MOTION DECOMPOSITION

2.1. Overview of our approach

Our goal is to obtain a hierarchical motion partition of the image represented by a tree delivering a view-based understanding of the dynamics of the scene. We call it the *motion decomposition tree* $\mathcal{T}$. This tree is composed of nodes which are associated with parametric motion models. In order to facilitate its estimation, we start with a compositional tree of candidate motion models attached to its nodes. We coin it the *proposal tree* $\mathcal{M}$, which is a superset of $\mathcal{T}$.

Given the color image pair (I_1, I_2), formed by two proximate frames of a video sequence, and a correspondence field $\mathbf{f}$ between the two, all defined on the image grid Ω, we are interested in recovering the tree $\mathcal{T}$ of motion models as a specific subtree of $\mathcal{M}$. The proposal tree $\mathcal{M}$ forms the search space of our problem. It contains a set of plausible compositional motion models that explain local evidence in different parts of the scene. We want to find the optimal decomposition tree $\mathcal{T}$ but also to associate a node to each pixel such that the visual motion at this pixel is approximated by the composition of the all motions on the path from the root to that particular node.

Each node $k \in \{1, \ldots, K\}$ of the proposal tree, numerated in level-order, is associated with a vector $\boldsymbol{\theta}_k$ of motion parameters defining a particular motion model. We may deal with different types of motion models depending on the nodes. Specifically, we will use a 8-parameter quadratic model for the root (0-th layer)and 6-parameter affine motion models for subsequent layers. For every pixel $p \in \Omega$, we define an index vector:

$$\boldsymbol{\alpha}(p) = (\alpha_1(p), \cdots, \alpha_k(p), \cdots, \alpha_K(p)), \qquad (1)$$

of K elements such that $\alpha_k(p) = 1$ if the pixel motion conforms to the parametric motion model given by $\boldsymbol{\theta}_k$, and $\alpha_k(p) = 0$ otherwise. Note that, $\alpha_k(p) = 1$ implies that $\alpha_{k'}(p) = 1$, for all nodes k' that are ancestors of k. Equivalently, if $\alpha_k(p) = 0$, then $\alpha_{k'}(p) = 0$ for all descendants of k. That is, the motion decomposition tree $\mathcal{T}$ is formed by the nodes of the larger tree $\mathcal{M}$ for which there is at least one pixel p such that $\alpha_k(p) = 1$.

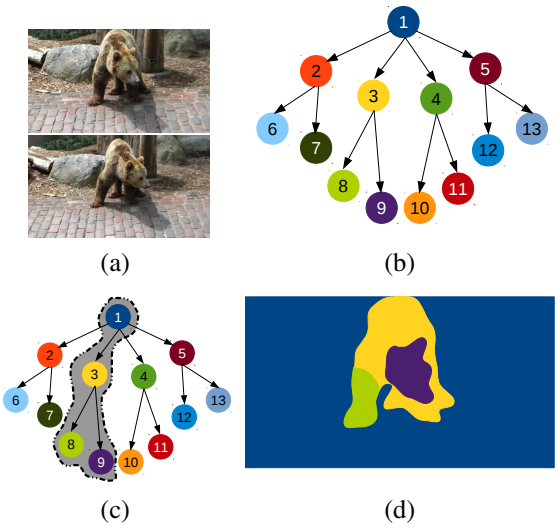


Fig. 2. Illustration of hierarchical motion decomposition on an image pair from the FMBS dataset [13]. (a) Input pair. (b) Initial proposal tree $\mathcal{M}$, with $K = 13$. (c) Decomposition tree $\mathcal{T}$ is emphasized with a gray mask. It corresponds to the selected node-labels from the proposal tree that better describe the dynamic scene (d). The per-pixel node-label assignment can be interpreted as a mid-level hierarchical motion segmentation of the scene. The colors of the segmentation result correspond to the colors of the nodes in the initial tree.

We group the K parameter vectors $\boldsymbol{\theta}_k$ in a collection $\mathbf{M} = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K\}$. We also define the displacement function $\mathcal{F}_p(\mathbf{M}, \boldsymbol{\alpha}(p))$ that results from composing the motion models indicated by $\boldsymbol{\alpha}(p)$ at p:

$$\mathcal{F}_p(\mathbf{M}, \boldsymbol{\alpha}(p)) = \sum_{k=1}^{K} \mathbf{w}_{\boldsymbol{\theta}_k}(p)\alpha_k(p), \qquad (2)$$

where $\mathbf{w}_{\boldsymbol{\theta}_k}(p)$ is the motion vector determined by motion model $\boldsymbol{\theta}_k$ at pixel p. We want compositional motion (2) at pixel p to agree with the reference correspondence field $\mathbf{f}(p)$.

An illustrative example can be found in Fig. 2. From the proposal tree $\mathcal{M}$ composed of nodes that describe distinctive motion models through the image pair, a subtree is selected to explain the motion of the different image parts. The background, for instance is clearly associated with the root node. The motion of the background is determined by the camera motion, however, the camera motion does not only affect this region, but the rest of the image as well. This is why, the body of the bear is associated to a child node of the root node. Thus, the motion model in this region is computed incrementally from its parent motion (Eq.2). Finally, one leg and the head of the bear exhibit movements of their own, which add up to the bear global motion. How these models are actually extracted, and how the label assignment is performed is explained hereafter.

Fig. 3. From left to right, windows are constructed from the root window by splitting it iteratively. The proposal tree follows the structure and color code of Fig.1.

2.2. Constructing proposal motion tree $\mathcal{M}$

This step consists in building the initial proposal tree, while estimating the parametric motion models associated to each of its nodes. We first perform a window sampling by hierarchically partitioning the image frame, as described in Fig. 3. It yields the desired tree structure, and each window will be the estimation support of the motion models attached to the corresponding nodes of the proposal tree.

The parametric motion models are computed robustly with the publicly available *Motion2D* software [14]. It is important to note here that the motion models describe per-region motion characteristics, in contrast to the instrumental motion field $\mathbf{f}$, which is an unstructured dense reference. Reasoning on piecewise parametric motion models have been successfully used recently to solve intricate problems like optical flow [15] and occlusion detection [16].

In order to compute the motion model corresponding to each window, the reference color image is compositely warped as we descend the tree. In this way, every motion model captures compositional aspects of the scene motion. The window sampling initialization procedure might seem somewhat arbitrary. Nevertheless, it provides an efficient means for the initial estimation of the motion models which will be further updated along with the determination of moving regions, as explained in the next session.

2.3. Estimating decomposition tree $\mathcal{T}$ and pixel labels

To recover the decomposition tree $\mathcal{T}$, we formulate the problem as a per-pixel label selection interleaved with motion models estimation. The labels represent the set of nodes from the proposal tree $\mathcal{M}$ which are selected to explain globally the input correspondence field $\mathbf{f}$. We introduce the collection of assignment vectors $\mathbf{A} = \{\boldsymbol{\alpha}(p)\}_{p\in\Omega}$, where $\boldsymbol{\alpha}(p)$ is given by Eq. 1. We want to estimate the elements of the collection of motion models $\mathbf{M}$ and the assignment vectors $\mathbf{A}$ by minimizing the following objective function:

$$E(\mathbf{M},\mathbf{A}) = \sum_{p\in\Omega} ||\mathbf{f}(p) - \mathcal{F}_p(\mathbf{M},\boldsymbol{\alpha}(p))||_2^2 + \lambda \sum_{\{p,q\}\subset\Omega} \mu(\boldsymbol{\alpha}(p),\boldsymbol{\alpha}(q)) \exp\left(-\frac{||p-q||^2}{2\beta_a^2} - \frac{||I_1(p)-I_1(q)||^2}{2\beta_b^2}\right), \quad (3)$$

where $\mathbf{f}$ is the reference flow field which is precomputed and used as input of our algorithm. We use *DeepFlow* [18] to compute it. The smoothness term in (3), where $\mu(\cdot,\cdot) = 1$ if the two input index vectors are different and 0 otherwise, is weighted by a positive parameter λ and by a joint spatial and color Gaussian kernel. The space and color parameters in this kernel, β_a and β_b, are hand-tuned. This pairwise energy term is computed over all possible pixel pairs. That is, our CRF model is fully connected. We have found that full connectivity allows our model to outline more accurately moving objects and moving objects parts.

This energy is minimized by performing coordinate descent on the collections $\mathbf{M}$ and $\mathbf{A}$, as presented in Alg. 1. On each iteration, $\mathbf{A}$ is updated by minimizing (3) with fixed $\mathbf{M}$ using an efficient message passing implementation based on the mean fields approximation and high dimensional filtering [17, 19]. In turn, the parametric motion models $\mathbf{M}$ are re-estimated with *Motion2D* at each step with $\mathbf{A}$ fixed. Recall that the motion models are compositional. Specifically, Eq. 2 is applied at every iteration step on the updated elements of the collection $\mathbf{M}$. Finally, the motion decomposition tree $\mathcal{T}$ is obtained by pruning from $\mathcal{M}$ all the nodes that are not selected by any of the elements of collection $\mathbf{A}$.

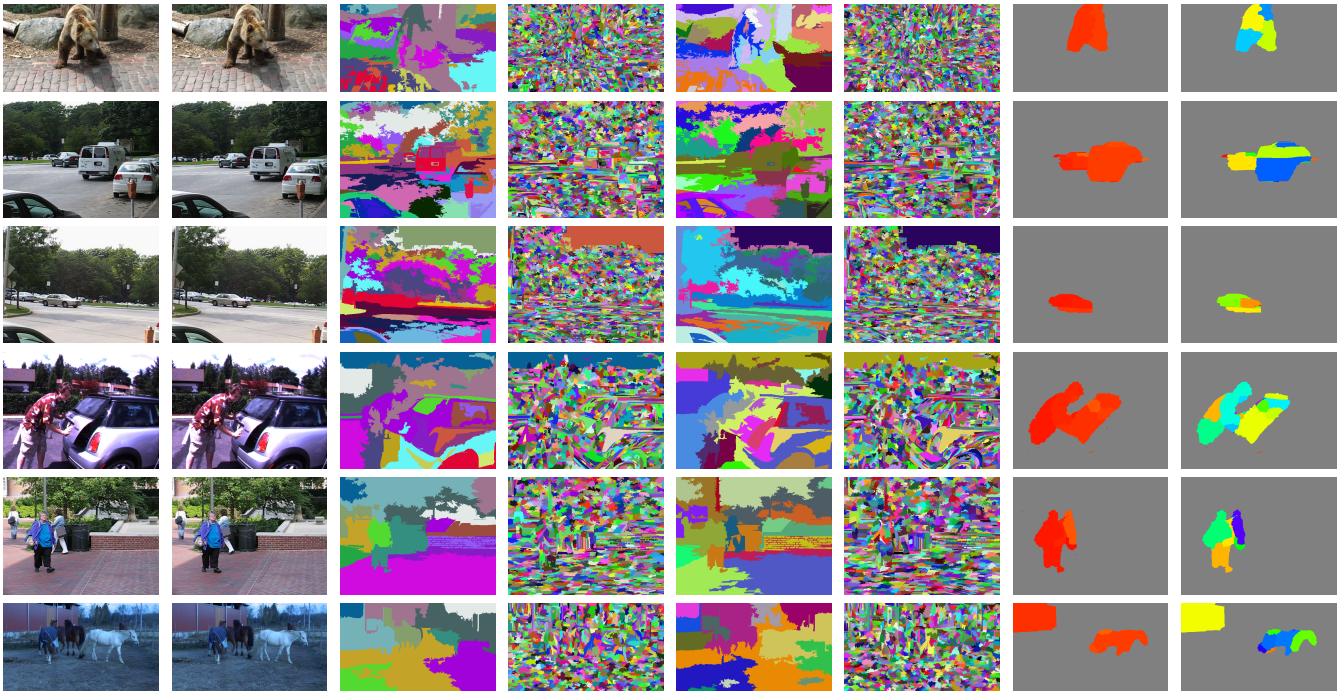
Algorithm 1 Motion decomposition tree estimation

procedure OPTIMIZATION($\mathbf{M}$, $\mathbf{A}$)
  $\mathbf{M} \leftarrow$ Compute proposal tree
  while `Not converged` **do** ▷ Or max. iterations
    $\mathbf{A} \leftarrow$ Minimize (3) w.r.t $\mathbf{M}$ with message passing [17]
    for $k = 1 \cdots K$ **do**
      if $\alpha_k(p) = 1$ `for some` $p \in \Omega$ **then**
        $\theta_k \leftarrow$ Update model with [14] for image warped with composed motion models of all ancestors of k.
  return $(\mathbf{M}, \mathbf{A})$

3. EXPERIMENTAL RESULTS

We report experimental comparative results with two popular hierarchical segmentation methods, *i.e.*, a graph-based segmentation method (GBH) [8], and a streamed hierarchical segmentation method (SHS) [7] (Fig. 5). We used the FMBS dataset [13], which provides sequences with a wide range of content and motion types. Before running our method, we apply iterative edge preserving filtering [20]. We tuned the parameters of the energy function (3) by hand and fixed them for all our experiments as follows. The proposal tree structure consists of one root node, eight nodes in the second layer, each of which subdivides in four nodes, each of which splits in two nodes ($K = 1 + 8 + 32 + 64 = 105$). The choice of these numbers was governed by our expectation of a maximum number of eight independent moving objects in the scene, and a limited subsequent decomposition into visible moving subparts, while bounding the size of the proposal

Fig. 4. Color code used to show our hierarchical segmentation results for a 3-deep tree where nodes in 0-th, first and second layer have respectively 8, 4 and 2 children each. Each node corresponds to a different color of the HSV colormap.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. Comparative visual results of scene motion segmentation. (a-b) Input image pair. (c-d) Segmentation results at the second and 15th layer of GBH [8]. (e-f) Segmentation results at the second and 15th layer for SHS [7]. (g-h) Segmentation results extracted by our motion decomposition algorithm for the second and 3rd layer. The color code used is indicated in Fig. 4.

tree. We set $\lambda = 10$, $\beta_a = 5$, and $\beta_b = 3$. For [7] and [8] we use the parameters proposed by the respective authors.

For an easy comparison, we only display segmentation results at first level (root children) and at the last level (leaves) of the hierarchy for all the presented methods. From Fig. 5, it can be observed that our method extracts insightful information from scenes which can contain a wide range of motion types. Furthermore, it is the only one to consistently provide segments that preserve some level of semantic meaning in terms of moving objects and moving parts, at both ends of the tree-based hierarchy. For instance, in Fig. 5(g) we can see that our method captures the main moving objects while also correctly labeling background pixels only with the root node. Moreover, our method is the one that suffers the least, by far, from over-segmentation as it is not driven only by local dissimilarity measures. One limitation of our method is its dependency to the reference optical flow method, which may be affected by adverse conditions as specular reflections (*e.g.,* second and last rows in Fig. 5).

4. CONCLUDING REMARKS

We have presented a method to compute a motion-based mid-level representation of a scene from a pair of images. It performs a hierarchical decomposition of the moving entities of the scene. Our method can be a valuable contribution to video analysis and editing pipelines. The motion decomposition can be interpreted as a hierarchical per-frame scene segmentation which captures relationships between different moving entities in the image. We have reported results on real image sequences that demonstrate the superior ability of our method to capture the main moving objects of the scene in the first layer of the tree, and to segment them in moving parts in deeper layers. As such, we believe our segmentation method is closer to the complex needs of video editing than current hierarchical segmentation approaches. Furthermore, our method may be used both as optimal search space and mid-level representation for action localization in video.

5. REFERENCES

[1] Juan-Manuel Pérez-Rúa, Tomas Crivelli, and Patrick Pérez, "Background-foreground tracking for video object segmentation," in *ICIP 2015, Quebec City*.

[2] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro, "Video snapcut: robust video object cutout using localized classifiers," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 70, 2009.

[3] Yong Jae Lee, Jaechul Kim, and Kristen Grauman, "Key-segments for video object segmentation," in *ICCV 2011, Barcelona*.

[4] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek, "Action localization with tubelets from motion," in *CVPR 2014, Columbus*.

[5] Chenliang Xu and Jason J Corso, "Evaluation of supervoxel methods for early video processing," in *CVPR 2012, Providence*.

[6] Jue Wang, Bo Thiesson, Yingqing Xu, and Michael Cohen, "Image and video segmentation by anisotropic kernel mean shift," in *ECCV 2004, Prague*.

[7] Chenliang Xu, Caiming Xiong, and Jason J Corso, "Streaming hierarchical video segmentation," in *ECCV 2012, Florence*.

[8] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR 2010, San Francisco*.

[9] Gunnar Johansson, "Visual perception of biological motion and a model for its analysis," *Attention, Perception, & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[10] Samuel J Gershman, Joshua B Tenenbaum, and Frank Jäkel, "Discovering hierarchical motion structure," *Vision Research*, 2015.

[11] Denis Fortun, Patrick Bouthemy, and Charles Kervrann, "Optical flow modeling and computation: a survey," *Computer Vision and Image Understanding*, vol. 134, pp. 1–21, 2015.

[12] Etienne Mémin and Patrick Pérez, "Hierarchical estimation and segmentation of dense motion fields," *International Journal of Computer Vision*, vol. 46, no. 2, pp. 129–155, 2002.

[13] Thomas Brox and Jitendra Malik, "Object segmentation by long term analysis of point trajectories," in *ECCV 2010, Heraklion*.

[14] Jean-Marc Odobez and Patrick Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, 1995.

[15] Jiaolong Yang and Hongdong Li, "Dense, accurate optical flow estimation with piecewise parametric model," in *CVPR 2015*.

[16] Juan-Manuel Pérez-Rúa, Tomas Crivelli, Patrick Bouthemy, and Patrick Pérez, "Determining occlusions from space and time image reconstructions," in *CVPR 2016, Las Vegas*.

[17] Philipp Krähenbühl and Vladlen Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *NIPS 2011, Granada*.

[18] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *ICCV 2013, Sidney*.

[19] Philipp Krähenbühl and Vladlen Koltun, "Parameter learning and convergent inference for dense random fields," in *ICML 2013, Atlanta*.

[20] Carlo Tomasi and Roberto Manduchi, "Bilateral filtering for gray and color images," in *ICCV 1998, Bombay*.